# Análisis de la semántica declarativa trivaluada $\mathcal{GS}$ para la programación en lógica rebatible básica

**Laura A. Cecchi**
Departamento de Informática y Estadística
UNIVERSIDAD NACIONAL DEL COMAHUE
e-mail: lcecchi@uncoma.edu.ar

**Guillermo R. Simari**
Departamento de Ciencias de la Computación
UNIVERSIDAD NACIONAL DEL SUR
e-mail: grs@cs.uns.edu.ar

**PALABRAS CLAVE:** Extensiones de la Programación en Lógica. Semántica Declarativa de Extensiones de la Programación en Lógica. Semántica de Juegos. Sistemas Argumentativos.

## Introducción

El uso de la Programación en Lógica (de ahora en más P.L.) como una herramienta de representación del conocimiento y del razonamiento de sentido común está basado en la idea de proveer a las máquinas con una especificación lógica del conocimiento, independiente de cualquier implementación, de tal modo que sea fácil de manipular y de razonar.

Consecuentemente, un significado preciso debe ser asociado con cualquier programa lógico con el objeto de proveer una especificación declarativa. La performance de cualquier mecanismo computacional es evaluada comparando su comportamiento con la especificación provista por la semántica declarativa. Encontrar una semántica declarativa adecuada para los programas lógicos es una de las áreas de investigación de la P.L. más importantes y difíciles

La Programación en Lógica Rebatible (de ahora en más P.L.R.) [Dun95,GS99,GSC98, Gar97], que es una extensión de la P.L., incorpora a los programas básicos un nuevo conjunto de reglas que permiten representar información tentativa, las *reglas rebatibles*. De este modo, un programa lógico rebatible consiste de un conjunto de dos clases de reglas: reglas *estrictas* y *rebatibles*. Las reglas estrictas son reglas en el sentido clásico de la programación en lógica: siempre que las premisas de una regla son satisfechas, tenemos permitido aplicarla y obtener la conclusión. Las reglas rebatibles son reglas que pueden ser derrotadas en presencia de evidencia contraria. Aunque la información tentativa representada puede ser eventualmente contradictoria, la P.L.R. provee un criterio para resolver los conflictos entre las reglas rebatibles con consecuentes contradictorios.

La semántica operacional de la P.L.R. se basa en un análisis dialéctico de argumentos y contraargumentos. Una consulta $q$ tendrá éxito si existe un argumento $\mathcal{A}$ de $q$ que lo justifique, *i.e.*, no existen contraargumentos que derroten a $\mathcal{A}$. Ya que los derrotadores son también argumentos podrían existir derrotadores para éstos últimos y así sucesivamente. El análisis dialéctico completo puede ser descripto a través del árbol dialéctico. La raíz del árbol dialéctico es un argumento para la consulta y todos los contraargumentos del nodo padre son considerados en cada nivel del árbol.

Si bien la semántica operacional es suficiente para verificar si una consulta está justificada o no en un programa lógico rebatible, creemos que el beneficio de una semántica declarativa para la P.L.R. es doble. Por un lado, es necesaria para ayudar al programador a especificar el conocimiento y razonar a partir de él sin preocuparse por la parte de control del sistema. Por otro lado, la definición de una semántica declarativa ayuda en el estudio de la P.L.R. como sistema de razonamiento no monótono. Un conjunto de propiedades [Dix95a, Dix95b, Dix95c], han sido presentadas con el objeto de clasificar y caracterizar el comportamiento de las semánticas de los programas lógicos con negación. Analizando el conjunto de propiedades que se cumplen es posible comparar al sistema con otros, mostrando sus ventajas y desventajas.

En [CS99, CS00] se introdujo la semántica declarativa trivaluada $\mathcal{GS}$, basada en la estructura de juegos, que permite modelar la semántica operacional de la P.L.R. presentada en [Gar97], en donde el criterio para decidir entre argumentos contradictorios no permite elementos incomparables.

El objetivo de este trabajo es presentar un análisis del conjunto de las consecuencias de un programa lógico rebatible. Como resultado inmediato surge un teorema en el que se prueba que la semántica $\mathcal{GS}$ es sensata y completa con respecto a la semántica operacional.

## La semántica declarativa trivaluada $\mathcal{GS}$

En [Abr97] y [AM97], se introduce a la semántica basada en juegos con el objeto de modelar a la computación como un juego entre dos participantes: el Sistema y el Ambiente. La idea presentada es la de utilizar un juego para modelar la *interacción* entre el Sistema y el Ambiente. Una "corrida" o "cómputo" simple involucra la interacción entre el proponente y el oponente y es representado por una secuencia de *movidas* hechas en forma alternada por *P* y *O*. Un juego especifica el conjunto de todas las posibles corridas o movidas legales y puede ser pensado como un árbol.

La justificación de un literal puede ser vista como un juego en donde un jugador propone un argumento para una meta $q$ mientras que el otro jugador, el oponente, trata de buscar contraargumentos que lo derroten. De este modo, podemos modelar al sistema de dialéctica a través de la interacción de dos jugadores : el proponente y el oponente.

La semántica $\mathcal{GS}$ se basa en la idea de construir la familia de todos los juegos para un literal $q$, donde cada juego comienza con un argumento diferente que soporta a $q$ . Informalmente, cada juego representa un árbol de dialéctica cuya raíz es un argumento a favor del literal $q$.

Una vez definida la familia de juegos, diremos que $q$ tendrá valor de verdad *verdadero* si en la familia existe un juego ganado por el proponente, *i.e.,* ganado por el jugador que propuso el primer argumento a favor de $q$. Diremos que el valor de verdad de $q$ es *falso*, si no existe en la familia ningún juego ganado por el proponente. De otro modo, la familia de juegos será vacía, *i.e.*, no existe ningún argumento que soporte a $q$ y por lo tanto no podremos jugar ningún juego. En este último caso, diremos que el valor de verdad de $q$ es *desconocido*.

Notaremos a la semántica $\mathcal{GS} = \langle V, F \rangle$. El conjunto de literales que pertenecen al lenguaje del programa lógico rebatible básico, y no pertenece a $V \cup F$, es el conjunto de literales cuyo valor de verdad es *desconocido*.

El valor de verdad de los literales depende del programa completo y no del valor de verdad del átomo correspondiente. Por ejemplo, podría darse el caso que bajo un programa lógico rebatible un literal $q$ tuviese valor de verdad *verdadero*, *i.e.*, existe un juego en la familia de juegos de $q$ ganado por el proponente, y que la familia de juegos de $\sim q$ fuese vacía en cuyo caso su valor de verdad es *desconocido*; o bien que la familia de juegos de $\sim q$ no contuviese ningún juego que fuese ganado por el proponente, en cuyo caso su valor de verdad es *falso*. La razón de que el valor de verdad de $\sim q$ no dependa del valor de verdad de $q$ está estrechamente ligada al hecho de que para cada literal $L$ analizamos su familia de juegos, sin tener en cuenta la familia de juegos del complemento de $L$.

Análogamente, el conjunto compuesto el complemento de todos los elementos en $F$ no necesariamente pertenecen a $V$. Por ejemplo, en el siguiente programa lógico $\Pi=\{d\}$, $\Delta=\{\sim a \prec b \wedge c ;\ b \prec c ;\ \sim b \prec d ;\ c \prec d\}$ bajo el criterio de especificidad (en este caso se comporta según las condiciones exigidas) $\sim a$ es *falso,* pues el único juego es ganado por el oponente, aunque $a$ no sea verdadero, ya que la familia de juegos es vacía.

Así el conjunto de consecuencias de un programa lógico rebatible coincide exactamente con el conjunto $V$ en $\mathcal{GS}$.

Un resultado interesante de la semántica $\mathcal{GS}$ es el siguiente. Por razones de espacio no se presenta la demostración.

> Teorema : Sea $\mathcal{P}$ un programa lógico rebatible donde la relación de preferencia entre contraargumentos es total, entonces
>
> $q$ es un literal justificado en P si y sólo si $q$ pertenece al conjunto $V$ de $\mathcal{GS}$

Este teorema indica que la semántica definida es sensata y completa con respecto a la semántica operacional definida para la programación en lógica rebatible, siempre y cuando la relación de preferencia entre contraargumentos sea total.

## Conclusiones y Trabajos Futuros

El principal aporte de este trabajo es la sensatez y completitud de la semántica trivaluada $GS$ con respecto a la semántica operacional, definida en función del árbol dialéctico. De este modo, hemos definido de un modo declarativo el conjunto de las consecuencias de tales sistemas.

Nuestro objetivo final es el de definir el conjunto de consecuencias de un programa lógico rebatible donde la relación de preferencia entre argumentos no esté restringida. En particular, estamos interesados en definir declarativamente el conjunto de consecuencias de los programas lógicos rebatibles cuando la preferencia entre argumentos sea la especificidad.

Entre nuestros trabajos futuros se encuentra también, extender la semántica para poder dar significado a programas lógicos rebatibles que incluyan la negación por falla.

## Referencias


[Abr97] Samson Abramsky. Semantics of Interacion. In A. Pitts. and P. Dibyer, editors, Semantics and Logic Computation. Cambridge, 1997.

[AM97] Samson Abramsky and Guy McCusker. Game Semantics. In *Proceedings of Marktoberdorf'97 - Summer School,* 1997.

[CS99] Laura A. Cecchi and Guillermo R. Simari. Game-based approach for modeling dialectical analysis. In *Proceedings of CACiC'99.*Tandil, 1999.

[CS00] Laura A. Cecchi and Guillermo R. Simari. Una semántica declarativa basada en juegos para la programación en lógica rebatible básica. Enviado a ICIE 2000.

[Dix95a] Jürgen Dix. A classification theory of semantics of normal logic programs : I. Strong properties *Fundamenta Informaticae*, XXII(3) :227-255, 1995.

[Dix95b] Jürgen Dix. A classification theory of semantics of normal logic programs : II. Weak properties *Fundamenta Informaticae*, XXII(3) :257-288, 1995.

[Dix95c] Jürgen Dix. Semantics of logic programs : Their intuitions and formal properties. An overview. In André Fuhrmann and Hans Rott, editors, *Logic, Action and Information*, páginas 227-312. Gruyter, 1995.

[Dun95] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. *Artificial Intelligence*, 77 :321-357,1995.

[GS99] A. García and G.R. Simari. Strong and Default Negation in Defeasible Logic Programming. In *4th Duth/German Workshop on Nonmonotonic Reasoning Techniques and their applications,* Amsterdam, 25-27, Marzo 1999.

[GSC98] A. García , G.R. Simari and Carlos Chesñevar. An Argumentative Framework for Reasoning with Inconsistent and Incomplete Information. In *ECAI 98, Proceedings of Workshop on Practical Reasoning and Rationality, 13th European Conference on Artificial Intelligence,* Brighton, England, *1998.*

[Gar97] A. García. La programación en lógica rebatible : su definición teórica y computacional. Tesis de Magister, *1997.*